拡張ユニットチュートリアル
（問題解決編）

TUT-0008　Ver. 1.1

Interface®
www.interface.co.jp

改訂履歴

Ver.	年　月	改　訂　内　容
1.0	2003年8月	新規作成
1.1	2005年9月	・誤記修正 ・技術資料更新

本チュートリアルをご使用の際は、必ず各製品型式の最新のドキュメント(ユーザーズマニュアル,オンラインヘルプ)をあわせて参照してください。また、最新のドライバソフトウェアをご使用ください。ユーザーズマニュアル，ドライバソフトウェアは弊社Web siteからダウンロードできます。(オンラインヘルプはドライバソフトウェアに含まれています)

商標/登録商標

本チュートリアルに掲載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

保障の内容と制限

株式会社インタフェースはドキュメント内の情報の正確さに万全を期しています。万一、誤記または誤植などがあった場合、株式会社インタフェースは予告無く改訂する場合があります。ドキュメントまたはドキュメント内の情報に起因するいかなる損害に対しても株式会社インタフェースは責任を負いません。
ドキュメント内の図や表は説明のためであり、ユーザ個別の応用事例により変化する場合があります。

著作権、知的所有権

株式会社インタフェースは本製品に含まれるおよび本製品に対する権利や知的所有権を保持しています。
本製品はコンピュータ ソフトウェア（プログラム）、 図、 文章、 写真などを含んでいます。

複製の禁止

株式会社インタフェースの許可なく、本製品(ドキュメント含む)の全て、または一部に関わらず、複製、改変などを行うことはできません。

責任の制限

株式会社インタフェースは、株式会社インタフェース または再販売者の予見の有無にかかわらず発生したいかなる特別損害、偶発的損害、間接的な損害、重大な損害について、責任を負いません。

―目　次―

はじめに 4

第 1 章　拡張ユニットとバスブリッジ 5

1.1　拡張ユニットって何？ 5
1.2　バスブリッジって何？ 6

第 2 章　拡張ユニットの原理と仕組み 7

2.1　コンピュータの電源を入れた時、内部で何が起きているのか？ 7
2.2　PCI デバイスの設定に使われる、PCI コンフィギュレーション空間とは何か？ 12
2.3　コンピュータはどうやって複数の拡張ボードを見分けているのか？ ..14
2.4　何を設定すれば、拡張ボードの I/O やメモリがアクセスできるのか？16
2.5　コンピュータに拡張ユニットを接続した時、内部でどんなことが起きるのか？ 19

第 3 章　拡張ユニットを組み込む 22

3.1　PCI バスブリッジの拡張ユニット組み込み例 22
3.2　CardBus バスブリッジの拡張ユニット組み込み例....24

第 4 章　拡張ユニットの FAQ 27

バスブリッジで使えない拡張ボードはあるのか？ 27
拡張ユニットを使用する際、BIOS に対して何か設定する必要はあるか？ 29
正常に認識されない時、どんな対処を行えば良い？ 32
正常に動作しない時、どんな対処を行えば良い？ 37
問題が発生した時、デバイス表示ツールで、どこを確認すれば良い？39
CardBus バスブリッジで、デバイスのリソースが割り当てられない 42
拡張ボードの割り込み番号が重複しているが、使用しても大丈夫か？44

用語集 46

技術資料紹介 47

参考文献 48

はじめに

平素は格別のご高配を賜り、厚くお礼申し上げます。本冊子は弊社拡張ユニット/バスブリッジ製品の技術的内容と運用時の問題解決について説明したものです。
運用過程での手助けになればと考えております。

★最新情報
弊社拡張ユニット/バスブリッジ製品の最新情報は、弊社 Web site より参照する事ができます。
弊社 Web site: http://www.interface.co.jp

■関連するチュートリアルの紹介

拡張ユニットに関するチュートリアルは、本書以外にも以下のチュートリアルをご用意しております。
併せて参照してください。

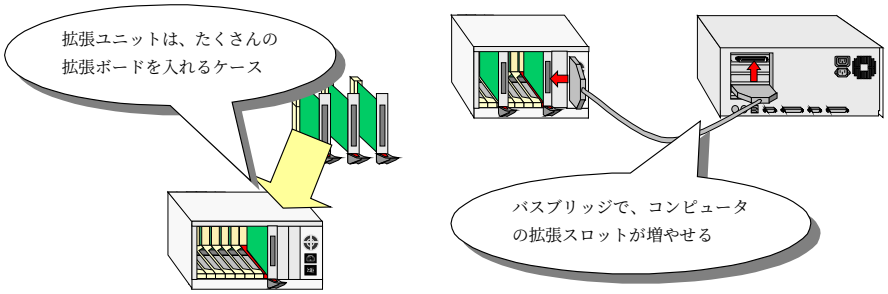
型式	名称
TUT-0001	弊社バスブリッジを使用したPCIバスによる Cバスボード制御 手順書
TUT-0002	弊社バスブリッジを使用したPCIバスによる ISAバスボード制御 手順書
TUT-0007	拡張ユニット チュートリアル(入門編)
TUT-0009	システムユニット チュートリアル

拡張ユニットの製品概要については、「拡張ユニット チュートリアル(入門編)」を併せて参照してください。
弊社CPUボードと組み合わせて使用するシステムユニットについての説明は、「システムユニット チュートリアル」を参照してください。

第1章　拡張ユニットとバスブリッジ

1.1　拡張ユニットって何？

拡張ユニットとは、複数の拡張ボードを納めることのできるケースのことを指します。バスブリッジと呼ばれる製品にて、コンピュータと接続することにより、コンピュータの拡張スロットの数を増やすことができます。



拡張ユニットは、単にコンピュータのスロットを増やすだけでなく、コンピュータに無い拡張ボードを使えるようにしたり、拡張ユニット内にCPUボードを組み込んで一体化させたりといった様々な用途に使用できます。

1.2 バスブリッジって何？

バスブリッジとは、コンピュータと拡張ユニットのバス(信号の束)を橋渡しする機能を持つものです。
このバスはローカルバスとも呼ばれ、CPUと外部デバイス(拡張ボードなど)との間を繋ぐ通信路を指します。

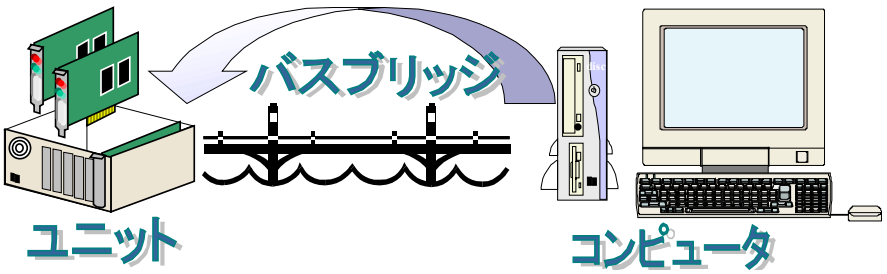


図 1-1 バスブリッジは2つのバスを繋ぐもの

コンピュータと拡張ユニットの間をバスブリッジで繋ぐことにより、コンピュータから使用できる拡張ボードの数を増やすことができます。

なお、本書では、特にPCIバスブリッジを扱います。
PCIバスは、現在コンピュータと外部デバイスを繋ぐ上で、汎用的かつ最速のローカルバスとして使用されています。

★その他のバス
PCI バス以外にも、幾つかバスの種類があります。

ISA バスや C(98)バスは、PCI バスが普及する以前、コンピュータのローカルバスとして使われていました。
AGP バスは、PCI バスを拡張して作られたバスで、グラフィックカードとの接続を考えられて作られたものです。
PCI Express は、PCI の次世代規格です。ソフトウェア面からは、上位互換を保っています。
CompactPCI や Low Profile PCI は、カードの形状は違いますが、PCI バスと同じものです。

なお、IEEE1394、USB は単なるシリアル通信であり、ローカルバスではありません(外部デバイスがコンピュータと直接アクセスすることができない)。これらは PCI バスの下に繋がる PCI デバイスを経由して使用されています。

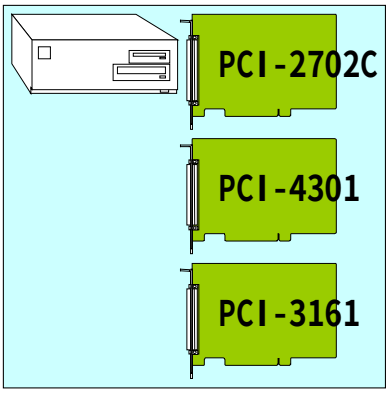
第2章　拡張ユニットの原理と仕組み

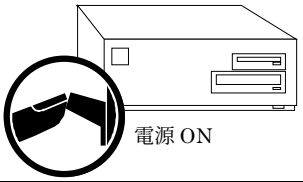
本章では、拡張ユニット製品を使用した際の動作原理と仕組みについて説明します。
これらを知ることにより、応用である使用方法を、より理解しやすくなります。

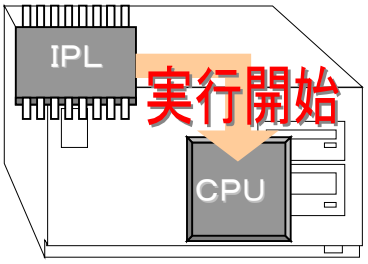
2.1 コンピュータの電源を入れた時、内部で何が起きているのか？

コンピュータの拡張スロットに拡張ボードを挿し、電源を入れた時、内部でどのようなことが起きているのか、事例を元に説明します。

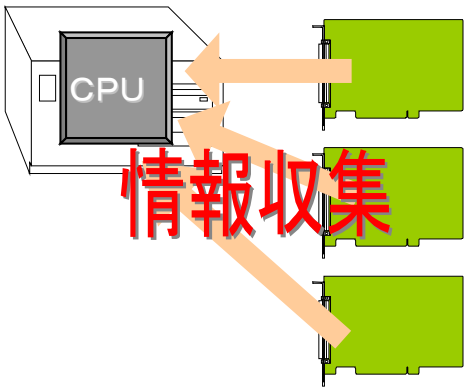
まず、コンピュータ内に、PCI-2702C，PCI-4301，PCI-3161の各1枚のPCIボードがセットされていることを想定します。



ステップ	解説
電源を入れる	コンピュータの電源を入れます。 コンピュータ内の拡張スロットに挿されている拡張ボードに、電源が供給されます。 

ステップ	解説
IPLの実行開始	電源が入るとコンピュータ内部では、自身の初期化を行うために、内蔵されたIPL(Initial Program Loader)と呼ばれる初期化プログラムの実行を開始します。  初期化作業として、CPUの初期化や、コンピュータ自身の構成確認、チェック作業が行われます。 IPLは、コンピュータ内に内蔵されたROMに格納されており、BIOS(Basic Input/Output System)とも呼ばれることがあります。
PnP初期化作業	IPLによる初期化作業中、コンピュータは自身に繋げられた各構成部品(ハードディスク，キーボード，拡張ボードなど)を認識／構成する作業に入ります。 この作業は、PCI拡張ボードなど、動的にI/Oポートやメモリアドレスに配置されるデバイスに対して、特に必要となります。 この設定作業を、Plug&Playと呼びます。 PC/AT互換機の仕様書などで、PnP BIOSと記述のあるものがありますが、これは、BIOSにPlug&Playを行う機能が内蔵されていることを明示しています。 ※PnP BIOSは、PCIスロットを持つPC/AT互換機であれば、ほぼ搭載しています。 ※上記以外、例えばSHやMIPSなどのCPUを搭載したCPUボードと呼ばれる製品は、PCIの構成を行わないものもあります。 ※弊社のSH4 CPUボード製品は、IPL時点ではPCIの構成を行いませんが、組み込まれているLinuxがOS起動時に行います。

<table>
<tr><th>ステップ</th><th>解説</th></tr>
<tr><td>PnP初期化作業：
必要リソースの取得作業</td><td>＜注意＞
Plug&Playの実際の実装処理は、BIOSごとに異なる場合があります。
ここでは、一般的な処理について説明します。

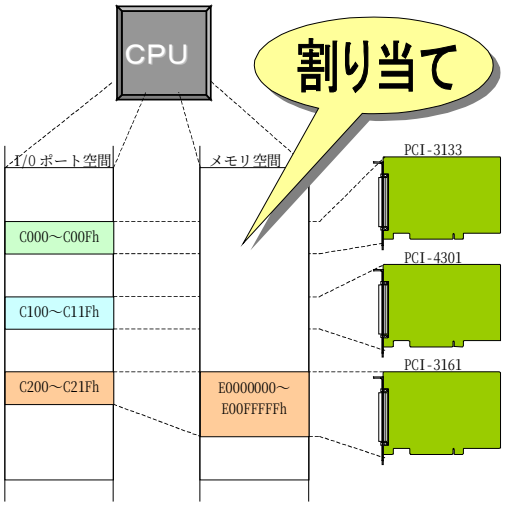
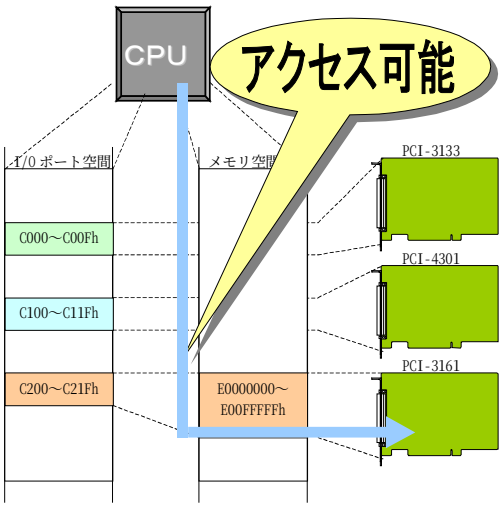
Plug&Play初期化作業において、コンピュータは各PCIデバイスを、I/Oポート，メモリアドレス，割り込み番号の、どこに割り当てるか決定するために、まず各PCIデバイスが、どんなハードウェアリソースを必要とするのか情報を収集します。
 

逆に言うと、この時点では、I/Oポート，メモリアドレスからPCIデバイスにアクセスすることは(リソースが割り当てられてないので)できません。
コンピュータがPCIデバイスにアクセスできる唯一の方法が、PCIコンフィギュレーションレジスタ空間へのアクセスです。

コンピュータは、PCIコンフィギュレーションレジスタ空間と呼ばれる専用の経路を介して、各PCIデバイスが必要とするハードウェアリソースを収集して回ります。</td></tr>
<tr><td>PnP初期化作業：
PCIボードの必要リソース</td><td>ここで、先の図に示したPCI拡張ボードを例に、コンピュータが収集したハードウェアリソースを示します。

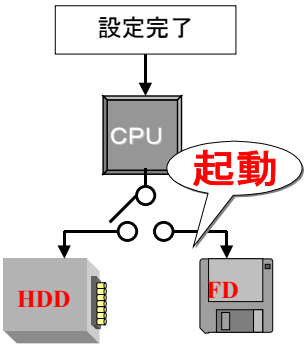
<table>
<tr><th>PCI デバイス</th><th>必要なリソース</th></tr>
<tr><td>PCI-2702C</td><td>I/O：16 ポート
割り込み：1 つ</td></tr>
<tr><td>PCI-4301</td><td>I/O：32 ポート
割り込み：1 つ
備考：バスマスタ使用</td></tr>
<tr><td>PCI-3161</td><td>I/O：32 ポート
メモリ：1MB</td></tr>
</table>

実際には、ここに示した以外に、複数のPCIデバイスが存在します。(グラフィックデバイスなどは、その代表的な例です)</td></tr>
</table>

ステップ	解説
PnP初期化作業： PCIデバイスへのリソース割り当て	収集した情報を元に、コンピュータは各PCIデバイスに対してハードウェアリソースを割り当てます。 PCIデバイスに対するハードウェアリソースの割り当ては、PCIコンフィギュレーション空間から行います。 コンピュータは、PCIコンフィギュレーション空間から、PCIデバイスのPCIコンフィギュレーションレジスタに然るべき値を書き込み、ハードウェアリソースの割り当てを行います。  ここで初めて、I/OポートやメモリアドレスからPCIデバイスにアクセスできるようになります。 逆に、ハードウェアリソースの割り当てが行われない間は、I/Oポートやメモリアドレスからは、PCIデバイスにアクセスすることができません。 

<table>
<tr><th>ステップ</th><th>解説</th></tr>
<tr><td>PnP初期化作業：
PCIボードのリソース設定完了</td><td>コンピュータは、PCI拡張ボードのハードウェアリソース設定を完了します。
例えば、以下のように設定しました。
<table><tr><th>PCI デバイス</th><th>必要なリソース</th></tr><tr><td>PCI-3133</td><td>I/O：C000～C00Fh
割り込み：IRQ11</td></tr><tr><td>PCI-4301</td><td>I/O：C100～C11Fh
割り込み：IRQ11</td></tr><tr><td>PCI-3161</td><td>I/O：C200～C21Fh</td></tr></table>
ここでI/OポートC000hにIN/OUTすると、PCI-3133のI/Oポートレジスタにアクセスすることができます。

なお、これらリソースの割り当ては、コンピュータの環境や状況によって変化します。</td></tr>
<tr><td>OSの起動</td><td>IPLによる初期化作業が終了すると、IPLはBIOSで設定された情報に従って、起動デバイスからブートしようとします。

BIOS設定画面で、起動デバイスをAドライブ(FDD)、Cドライブ(HDD)にするか設定を行いますが、これらの設定はコンピュータ内部のEEPROMに情報が保持されており、IPLがこの設定値を読み取って、然るべき起動デバイスから起動を行います。
 
</td></tr>
<tr><td>OSによるPnPの再設定</td><td>幾つかのOSでは、OS起動時、BIOSが設定したPCIデバイスのハードウェアリソースの割り当てを破棄し、再設定を行うものがあります。
これは、OS実行中にPCIデバイスの接続&切り離しやハードウェアリソースの移動をサポートするのにOS自身がPCIデバイスの全情報を管理する必要があるため、またはBIOSが処理し切れない設定を、OSが責任を持って行うためにあります。
OS自身がPCIデバイスの管理を行う代表例としては、Windows98やWindows2000／XPなどのマイクロソフト社のWindows関連製品、そしてオープンソースとして有名なLinuxもこれにあたります。
逆に、BIOSの設定した情報のまま、OS自身では再構成を行わないものとしては、MS-DOSやPC DOS，Windows NT 4.0があたります。</td></tr>
</table>

2.2 PCIデバイスの設定に使われる、PCIコンフィギュレーション空間とは何か？

先の説明で、PCIデバイスのハードウェアリソースの設定には、PCIコンフィギュレーション空間を経由し、PCIデバイスのPCIコンフィギュレーションレジスタに値を書き込むことで行う、と説明しました。

このPCIコンフィギュレーション空間とは何かと言うと、PCIデバイスのハードウェアリソースや固有の設定を行うための専用の領域です。

ISAやC(98)バスの拡張ボードでは、I/Oポートアドレスやメモリアドレス、割り込み番号を、拡張ボード上のRSW(ロータリスイッチ)やDSW(ディップスイッチ)、JP(ジャンパ)などで行っていました。
つまり、これらの諸設定は、人間が手で行っていました。が、PCIバス(CompactPCIも同様)の拡張ボードでは、これらの設定はソフトウェアで行われます。

項目	ISA	PCI
Plug&Play	無し※1	有り
アドレス幅	24bit	32/64bit
データ幅	16/8bit	32/64bit
データ転送速度	8Mbytes/sec	132Mbytes/sec(32bit) 264Mbytes/sec(64bit)

※1：Plug&Play ISA が規格化されたが、広まらなかった。

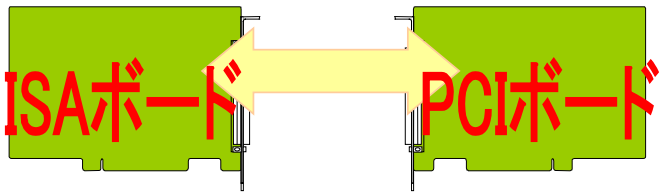


図 2-1 PCIとISAとの比較

各デバイスのI/Oポートアドレスやメモリアドレス，割り込み番号は、ソフトウェアから設定できるよう専用のレジスタを用意し、そこに書かれた値に従って動作します。
この専用のレジスタを、PCIコンフィギュレーションレジスタと呼び、PCIデバイスごとに最大256バイトまで保持できます。

I/Oポートにアクセスするには I/Oポート空間、メモリにアクセスするにはメモリアドレスを経由してアクセスしますが、PCIコンフィギュレーションレジスタにアクセスするには、このPCIコンフィギュレーション空間を経由してアクセスします。I/Oやメモリと異なるのは、アクセス位置の指定方法であり、I/OがC000h番地という具合に1つのパラメータで指定するのに対して、PCIコンフィギュレーション空間では、「バス番号:0，デバイス番号:5，ファンクション番号:0，レジスタ番号:4」という具合に、4つのパラメータを使って位置を特定します。

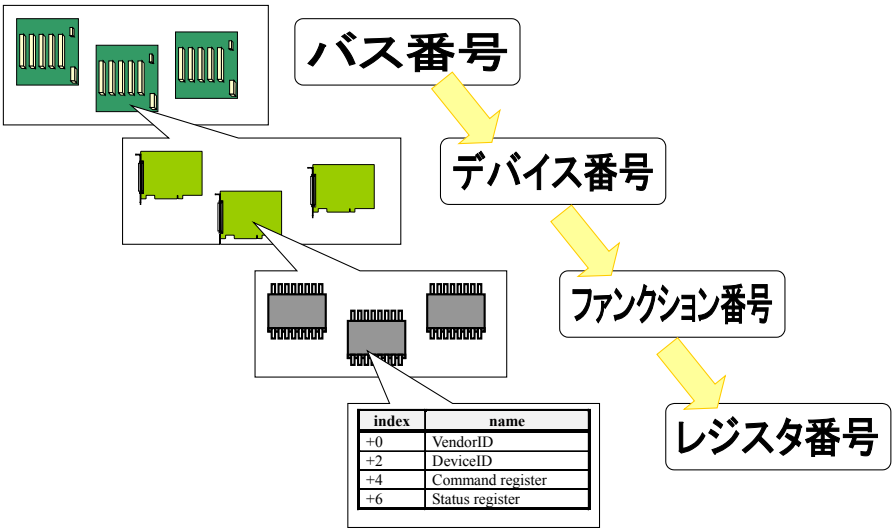


図 2-2 PCIは4つのパラメータで位置を指定する

BIOSやWindowsなどのOSが、PCIデバイスのハードウェアリソースを設定する時、このPCIコンフィギュレーション空間から個々のPCIデバイスのPCIコンフィギュレーションレジスタにアクセスし、デバイスが必要とするI/O、メモリ、割り込みの情報を得、収集した情報を元に設定を行っているのです。

2.3 コンピュータはどうやって複数の拡張ボードを見分けているのか？

例えば、コンピュータに拡張ボードを挿しWindowsを起動した時、OSがシステムに挿した新しい拡張ボードを認識し、ユーザに拡張ボードに対応したデバイスドライバの用意を促すよう指示を出します。

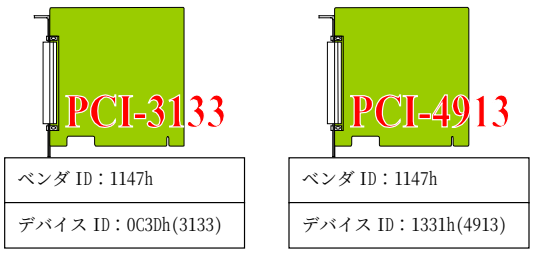
コンピュータは、どうやって新しい拡張ボードを認識できるのでしょうか？
答えは、先のPCIコンフィギュレーションレジスタにあります。

『2.2 PCIデバイスの設定に使われる、PCIコンフィギュレーション空間とは何か？』にて、PCIデバイスにはI/Oポートアドレスやメモリアドレスなどを設定するために、PCIコンフィギュレーションレジスタというものが存在すると説明しました。

規格では、PCIコンフィギュレーションレジスタの内、先頭64バイトがヘッダ領域と呼ばれ、規格で定められたフォーマットに従って構成するよう定義付けられています。
この内、先頭4バイトは以下に定義付けられています。

オフセット	レジスタ名	説明
+0h	ベンダID	各ベンダに割り当てられた一意の値を入れる。 これは、デバイスのメーカ名を識別するために使用する。
+2h	デバイスID	ベンダによって割り当てられたPCIデバイスの識別番号を入れる。
---	---	---

ここで弊社製品のPCI-3133とPCI-4913のベンダIDとデバイスID値を紹介します。

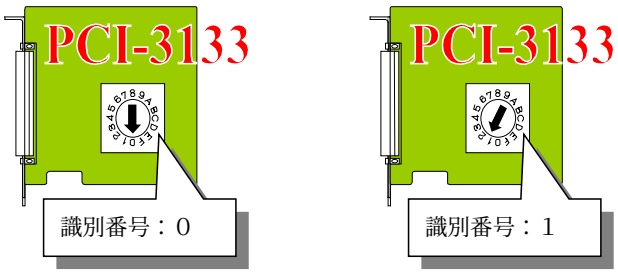


PCI-3133とPCI-4913のデバイスID値が異なります。WindowsなどのOSは、このベンダIDとデバイスIDの値を見て、拡張ボードの違いを認識します。
(実際には、デバイスを区別する一意のレジスタ。例えば、サブシステムIDとサブシステムベンダIDというレジスタなども合わせて使用されます)

★同一の拡張ボードが挿しこまれた時、どうやって区別する？
同一の拡張ボード、例えば PCI-3133 が 2 枚コンピュータに挿された場合、コンピュータは同一のデバイスが 2 つ存在することを認識できますが、それ以上の区別は基本的にできません。

つまり、A の処理をさせる PCI-3133 と B の処理をさせる PCI-3133 が在った時、ベンダ ID やデバイス ID などの情報から、両者を区別する(例えば A の処理を行う PCI-3133 はどちらか？)ことは難しいということです。

これに対処するため、弊社の PCI 関連製品では、拡張ボードの基板上にボード識別用の RSW を用意しております。
A の処理を行う PCI-3133 に識別番号 0、B の処理を行う PCI-3133 に識別番号 1 を設定することで、両者を確実に区別できるようにしております。



2.4 何を設定すれば、拡張ボードのI/Oやメモリがアクセスできるのか？

『2.1 コンピュータの電源を入れたとき、内部で何が起きているのか？』にて、拡張ボードのリソース割り当ては、PCIコンフィギュレーションレジスタに対して然るべき値を設定することで行うと説明しました。

何を設定することで、拡張ボードのI/Oポートやメモリアドレスにアクセスできるようになるのでしょうか？
答えは、先のPCIコンフィギュレーションレジスタにあります。

以下に、リソース割り当てに用いられる主要なレジスタを紹介します。

オフセット	レジスタ名	説明
…	…	…
+4	コマンドレジスタ	デバイスによるPCIバスアクセスの実行、および応答の可否等を決定する。
…	…	…
+10～24h	ベースアドレスレジスタ	PCIデバイスのI/Oおよびメモリ機能を、動的にシステムのI/Oおよびメモリアドレス空間にマッピングする。
…	…	…
+3Ch	割り込みレジスタ	PCIデバイスの割り込み線が、システム割り込みコントローラのどの割り込み要求信号線に接続しているかを識別する。
…	…	…

上表の内、ベースアドレスレジスタと割り込みレジスタは説明の通りで、それぞれベースアドレスレジスタはI/Oポートアドレスあるいはメモリアドレスの設定を、割り込みレジスタは割り込み番号を設定するレジスタです。

例えば、PCI-4913では、ベースアドレスと割り込み線は、次のように定義されています。

オフセット	レジスタ名	説明
+10h	I/Oポートアドレス	PCI-4913のI/Oポートアドレスを設定する。
+14h	メモリアドレス	PCI-4913のメモリアドレスを設定する。
+3Ch	割り込み	PCI-4913の割り込み番号を設定する。

例えば、+10hのI/Oポートアドレスレジスタの値をD000hに設定すると、PCI-4913のI/OポートアドレスはD000hに設定されます。

しかし、この値を設定しただけでは、I/OポートアドレスからPCI-4913のI/Oポートにアクセスすることはできません。
アクセス可否の鍵を握っているのが、コマンドレジスタです。

ここで、コマンドレジスタの定義(一部)を紹介します。

ビット	説明
0	I/O 空間制御 0:デバイスによる I/O 空間アクセスへの応答を禁止します。 1:デバイスによる I/O 空間アクセスへの応答を許可します。
1	メモリ空間制御 0:デバイスによるメモリ空間アクセスへの応答を禁止します。 1:デバイスによるメモリ空間アクセスへの応答を許可します。
2	バスマスタ制御 0:デバイスの PCI バスマスタアクセスを禁止します。 1:デバイスの PCI バスマスタアクセスを許可します。
…	…

コマンドレジスタは、I/Oポートおよびメモリを持ったPCIデバイスのアドレス空間に出現するためのスイッチの役割を持ちます。

例えば、PCI-4913が実際にI/Oポートアドレス空間のD000h上に現れるには、コマンドレジスタのビット0を1に設定する必要があります。
逆に、ビット0が0に設定されている場合、PCI-4913のI/OポートがI/Oポートアドレス空間に出現することはありません。

以上、コマンドレジスタとベースアドレスレジスタを設定することにより、拡張ボードのI/Oやメモリにアクセスできるようになります。

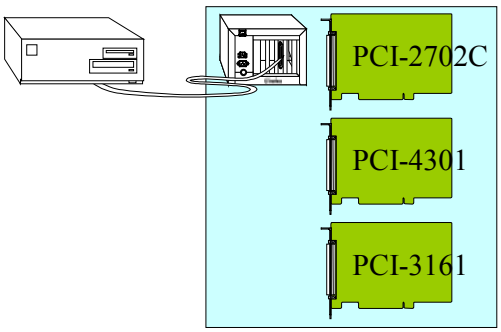
★バスマスタ制御
コマンドレジスタのビット 2 は、バスマスタ転送機能を持った PCI デバイスに対して、PCI バスマスタ転送の許可を与えるビットです。このビットを 1 に設定することで、PCI デバイスは PCI バスマスタ転送を行うことができます。
バスマスタ転送機能を持つ弊社製品としては、PCI-4302 や PCI-3176 などがあります。

★PCI バスマスタ転送機能
PCI バスマスタ転送機能は、ホスト PCI ブリッジの代わりに、PCI デバイスがバスのマスタとなる機能です。これにより、CPU を介さず PCI デバイスから直接コンピュータ上のメモリとやり取りできるため、CPU に負荷をかけず、非常に高速なデータのやり取りが実現できます。

ISA や C(98)バスの拡張ボードでは、DMA 転送と呼ばれる DMA コントローラが CPU を介さずデータ転送をやり取りする方式がありました。
PCI バスマスタ転送では、コンピュータ上の DMA コントローラの助けを借りず、PCI デバイス自身がやり取りする違いがありますが、目的は同じ高速なデータ転送にあります。

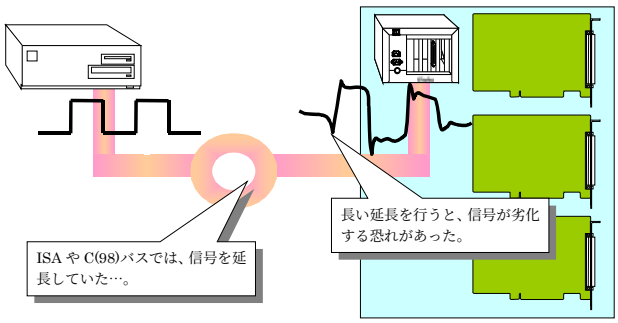
2.5 コンピュータに拡張ユニットを接続した時、内部でどんなことが起きるのか？

『2.1 コンピュータの電源を入れたとき、内部で何が起きているのか？』では、コンピュータの拡張スロットに拡張ボードを挿した際の、内部の動きについて説明しました。
では、コンピュータに拡張ユニットを接続し、拡張ユニットの拡張スロットに拡張ボードを挿して電源を入れた時、先の動きと何か違うことが発生するのでしょうか？



結論から言うと、内部の動きに関しては ほとんど違いがありません。
ただ、唯一異なるのは、コンピュータと拡張ユニット内の拡張ボードがPCI-to-PCIブリッジと呼ばれるデバイスを介して繋がるという点です。

この接続方法は、ISAやC(98)バスでは大きく異なります。
ISAやC(98)バス用の拡張ユニットは、バスの信号を単に延長しているだけでした。これは機構的に単純な反面、延長することで信号の品質が劣化しやすく、離れた場所にある拡張ボードが正常に動作しなくなる恐れがありました。
(弊社の拡張ユニット製品では、それらを抑えるために、様々な工夫を行っています)



単に信号を延長する方式では、信号が劣化しやすく、バスの伝送速度を上げることがままなりません。PCIバスでは、高品質かつ高速なデータ転送を実現するため、PCIデバイス同士を接続できる数を一定に限定し、信号の劣化を防ぎます。
PCIデバイスが集合する単位をバスと呼び、バスとバス同士はPCI-to-PCIブリッジと呼ばれるデバイスによって連結される仕組みです。

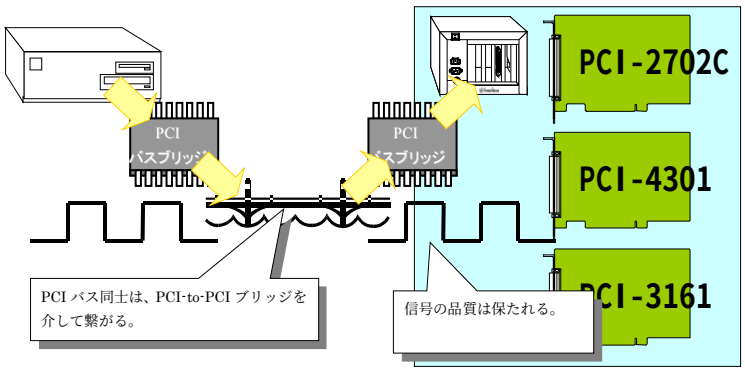


図 2-3 PCIバス同士はバスブリッジによって繋がる

注意が必要なのは、PCI-to-PCIブリッジで繋げた先のバスへのアクセスは、PCI-to-PCIブリッジが仲介して行うという点です。従って、PCI-to-PCIブリッジに正しい設定が行われないと、繋げた先のバスのPCIデバイスを正しくアクセスすることができません。

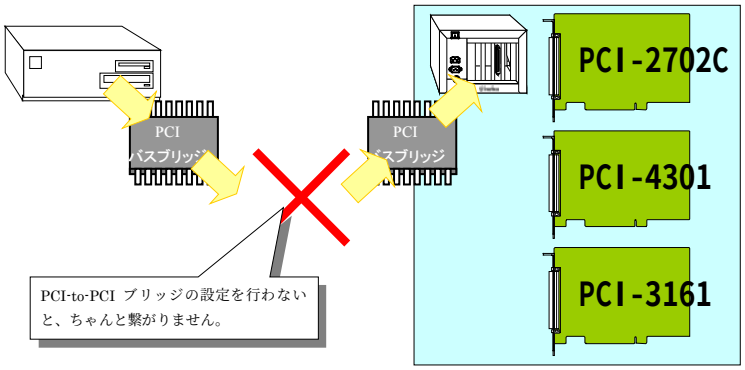


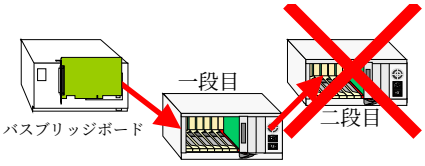
図 2-4 バスブリッジを設定しないとPCIバス同士は繋がらない

つまり、PCI-PCIバスブリッジは、ハードウェアとソフトウェアとの連携によって実現されていることがわかります。
BIOSやWindowsなどのOSは、これらPCI-PCIバスブリッジの設定を含めて高度な設定を内部で行っているのです。

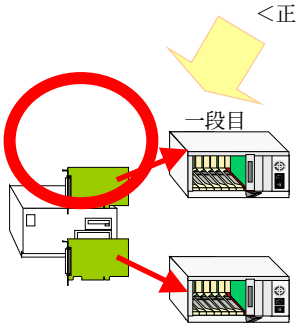
★バスブリッジの多段接続について
拡張ユニットをより多く使うため、バスブリッジを多段に接続したい場合があるかも知れません。
弊社では、以下の理由から 2 段以上の接続は未サポートとさせて頂いております。

理由：
多段接続した際、正常に動作しない場合があるのを幾つか確認しているためです。
これは、理論的に接続できないのでは無く、PCI-PCI 間の接続はハードウェアとソフトウェアとの連携により成り立っているため、使用する OS や BIOS によってハードウェアリソースを割り当てられないケースがあるからです。

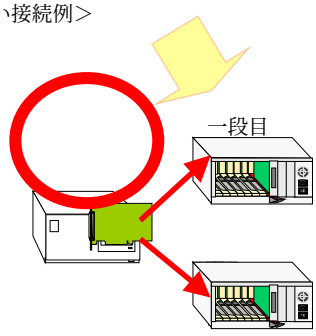
対策：
複数の拡張ユニットを使用するには、ホスト側コンピュータに使用台数のバスブリッジボードを挿して 1 段の構成で使います。
あるいは高速シリアルバスブリッジを使用することで、バスブリッジボード 1 枚に対して最大 2 台の拡張ユニットを接続することができます。



＜正しくない接続例＞



＜通常のバスブリッジの接続例＞



＜高速バスブリッジの接続例＞

第3章　拡張ユニットを組み込む

バスブリッジ製品の組み込みは、添付あるいはWeb siteで公開しておりますマニュアルを参照してください。

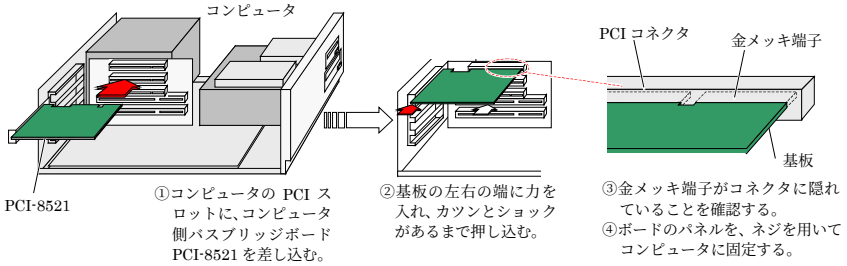
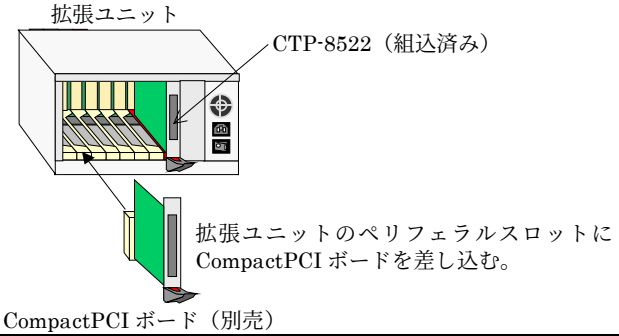
本章では、主な拡張ユニット製品について、マニュアルを元に、拡張ユニットの組み込み例を紹介します。

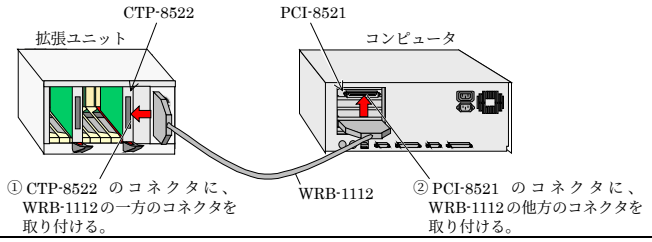
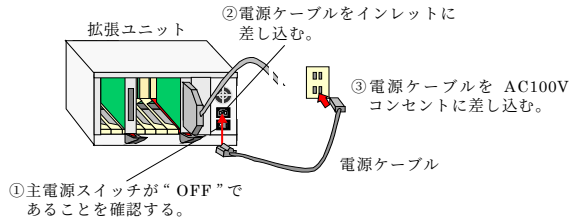
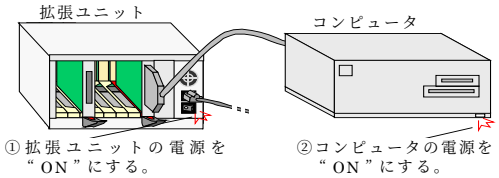
⚠注意

	最新情報を確認する際は、弊社Web siteを参照してください。 **弊社Web siteで公開しておりますマニュアルは、内容が常に最新に保たれています。** **本章は、参考事例およびセットアップマニュアルの補足事項について中心に説明します。**

3.1　PCIバスブリッジの拡張ユニット組み込み例

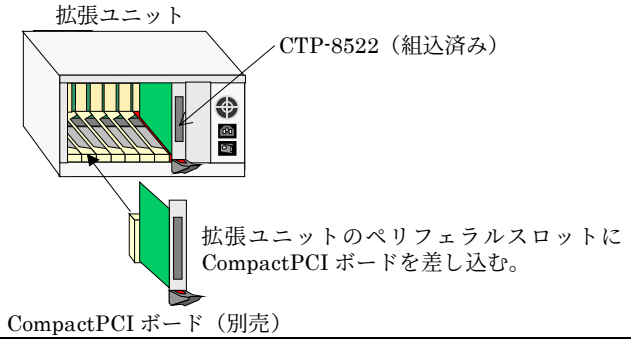
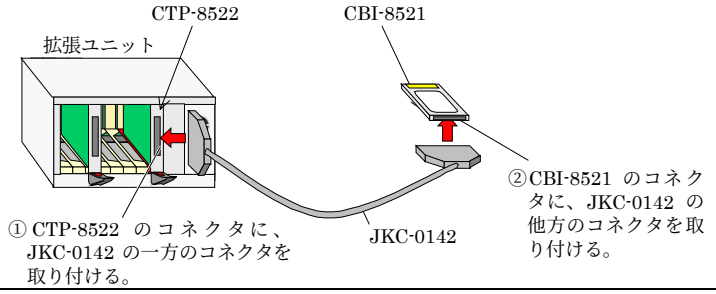
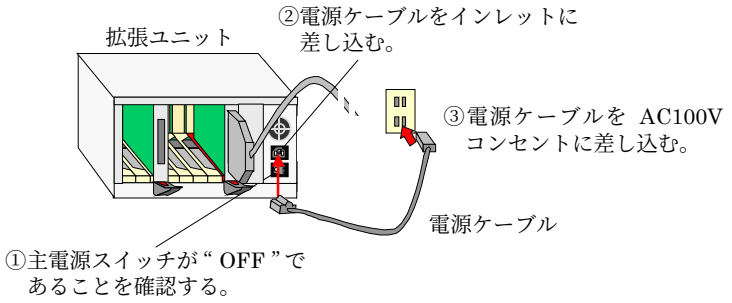
ここでは例として、PCI-CTU07DJを例に、PCIバスブリッジ付き拡張ユニット製品の組み込み例を紹介します。

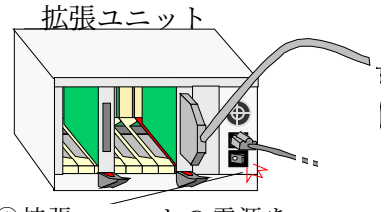
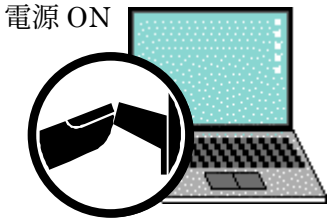
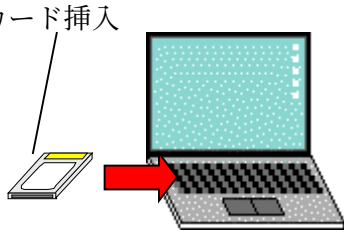
作業	詳細
コンピュータ側バスブリッジボードの実装	コンピュータのカバーを取り外した後、下図のように、コンピュータにコンピュータ側バスブリッジボードを実装します。  実装後、カバーを元に戻します。
拡張ユニットへのCompactPCIボードの実装	拡張ユニットのペリフェラルスロット(1～7)に、使用するCompactPCIボードを実装します。 

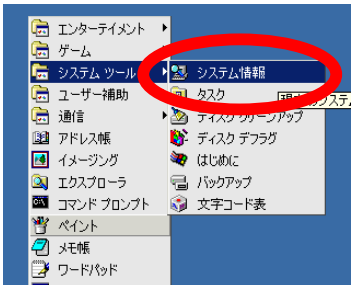
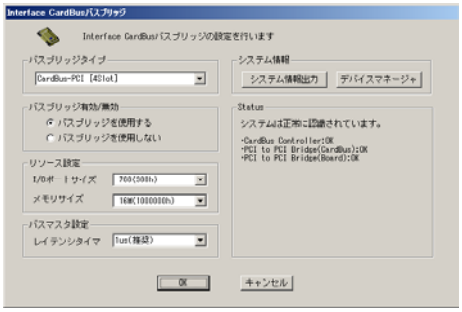
作業	詳細
ケーブル接続	下図のように、CompactPCI拡張ユニットとコンピュータを接続します。 
電源ケーブルの接続	下図のように電源ケーブルを接続します。 
拡張ユニットおよびコンピュータの起動	電源連動制御機能を使用せず、システムを起動する場合は、以下の手順で行ってください。  電源連動制御機能を使用する場合は、コンピュータの電源をONすると、拡張ユニットの電源もONになります。
初回起動時の操作	拡張ユニットにCompactPCIボードを追加して、初めてOSが起動されるときは、各ボードのドライバのインストールを行う必要があります。 以下では、Windows2000を例に、操作例を示します。 1.Windowsのデスクトップが表示されます。ネットワークなどへのログインダイアログが表示されているときは、ログインしてください。 2.拡張ユニットに挿入したボードのドライバ等をインストールしてください。各ボードのドライバのインストール方法については、それぞれのボード／ドライバ等の取扱説明書を参照してください。 3.ドライバによっては、「再起動する必要があります。」と表示される場合があります。「OK」を押して再起動してください。 4.必要に応じて、アクセサリの「システムツール」－「システム情報」の「ハードウェアリソース」で、各ボードの状態をご確認ください。
セットアップの完了	以上のセットアップで、ユニット上のCompactPCIボードを制御できるようになりました。 ボードに付属するドライバソフトウェアや、制御アプリケーションを用いてCompactPCIボードを制御できます。

3.2 CardBusバスブリッジの拡張ユニット組み込み例

ここでは例として、CBI-CTU07DJを例に、CardBusバスブリッジ付き拡張ユニット製品の組み込み例を紹介します。

作業	詳細
拡張ユニットへのCompactPCIボードの実装	拡張ユニットのペリフェラルスロット(1～7)に、使用するCompactPCIボードを実装します。 
ケーブル接続	下図のように、CompactPCI拡張ユニットとCardBusカード(CBI-8521)を接続します。 
電源ケーブルの接続	下図のように電源ケーブルを接続します。 

作業	詳細
拡張ユニットの電源ON	電源連動制御機能を使用せず、システムを起動する場合は、拡張ユニットの電源を先にONにしてください。  ①拡張ユニットの電源を“ON”にする。 電源連動制御機能を使用する場合は、コンピュータの電源をONすると、拡張ユニットの電源もONになります。
ノートパソコンの起動	ノートパソコンの電源をONにし、OSを起動します。 ここでは例として、Windowsを起動します。 
CardBus カード(CBI-8521)の挿入	Windowsが起動したら、ノートパソコンのCardBusスロットに、CardBusカード(CBI-8521)を挿入します。 
バスブリッジのドライバのインストール	バスブリッジドライバをインストールするようシステムから促されます。画面の指示に従いインストールを行うと、コントロールパネルに「Interface CardBusバスブリッジ」が追加されます。

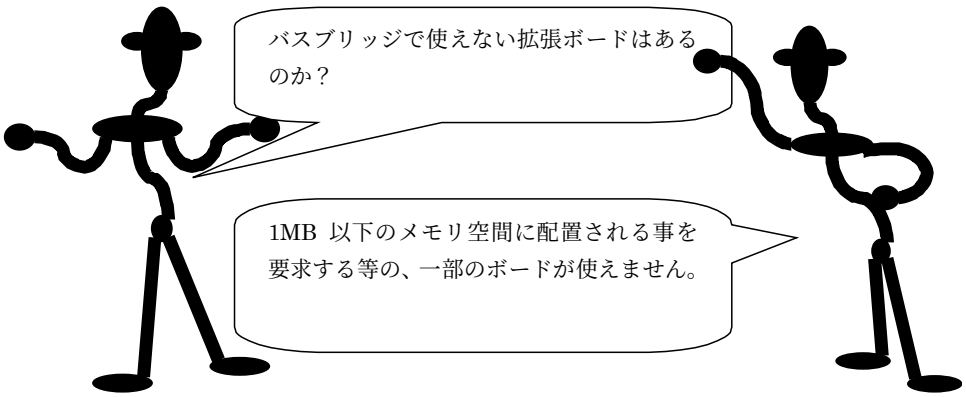
作業	詳細
拡張ボードのドライバのインストール	次に、拡張ユニットに挿した拡張ボードのデバイスドライバをインストールするようシステムから指示されます。 ①各ボードのドライバのインストール方法については、それぞれのボード／ドライバ等の取扱説明書を参照してください。 ②ドライバによっては、「再起動する必要があります。」と表示される場合があります。「OK」を押して再起動してください。 ③必要に応じて、アクセサリの「システムツール」－「システム情報」の「ハードウェアリソース」で、各ボードの状態をご確認ください。 
コントロールパネルの設定	使用する前に、コントロールパネルの「Interface CardBusバスブリッジ」の設定を行います。  設定時、注意する個所は以下の通りです。 ・バスブリッジタイプを使用する拡張ユニットに合わせる 使用する拡張ユニットのスロット数に合わせてバスブリッジタイプを設定します。 ・「バスブリッジを使用する」に選択する バスブリッジを使用する／しないという項目があるので、使用するに選択します。 その他の「リソース設定」と「レイテンシタイマ」項目は、問題が発生した時や高度なチューニングを必要とする時 使用します。 通常時、これらの項目を設定する必要はありません。
セットアップの完了	以上のセットアップで、ユニット上のCompactPCIボードを制御できるようになりました。 ボードに付属するドライバソフトウェアや、制御アプリケーションを用いてCompactPCIボードを制御できます。

第4章　拡張ユニットのFAQ

本章では、お客様から弊社技術サポートに寄せられた、拡張ユニット使用時につまづいた点や、ご質問について、一問一答形式で説明致します。

項目
バスブリッジで使えない拡張ボードはあるのか？
拡張ユニットを使用する際、BIOSに対して何か設定する必要はあるか？
正常に認識されない時、どんな対処を行えば良い？
正常に動作しない時、どんな対処を行えば良い？
問題が発生した時、デバイス表示ツールで、どこを確認すれば良い？
CardBusバスブリッジで、デバイスのリソースが割り当てられない。
拡張ボードの割り込み番号が重複しているが、使用しても大丈夫か？

バスブリッジで使えない拡張ボードはあるのか？



コンピュータの拡張スロットに直接拡張ボードを挿して使用する場合と、バスブリッジと介して拡張ユニットに挿した拡張ボードを使用する場合、基本的に差は生じません。
しかし、使用しているPCI-to-PCIブリッジの制限により、一部使えない製品があります。

例えば、以下の製品が使えません。

使えない製品	理由
PCIキャッシュ制御信号を利用する拡張ボード	PCIボードの中には、プリフェッチ可能メモリを使用するものがあります。 弊社で使用するPCI-to-PCIブリッジは、プリフェッチ可能メモリを正しく制御できないので、この機能を使用するPCIボードでは使用できません。
PCI to PCIブリッジ規格に対応していない拡張ボード	PCI-to-PCIブリッジ規格に対応していない拡張ボードは、PCI-to-PCIブリッジを経由してのアクセスはできません。
バス番号0上でのみ動作する拡張ボード	アクセス速度や使用目的から、バス番号0上でのみ動作するよう設計された拡張ボードがあります。 これらの代表的なものとしては、グラフィックアクセラレータボードなどがあります。
メモリアドレスの1MB空間以下に配置するよう、要求する拡張ボード	ほとんどのコンピュータでは、PCI-to-PCI ブリッジより先に、1MB 以下のアドレス情報が渡されません。 結果、1MB以下のメモリ空間に配置するよう強制するPCIボードは使用できません。

※上表は代表的な事例です。詳しい説明は、拡張ユニットやバスブリッジのマニュアルを参照してください。

なお、先の表に挙げた事項は、PCI-ISAおよびPCI-C(98)バスブリッジボードを除き(その他、PCI/ISA混載タイプも含みます)、弊社拡張ボードには当てはまりません。

弊社製品をご利用いただく限り、安心して拡張ユニットと拡張ボードをご使用いただけます。

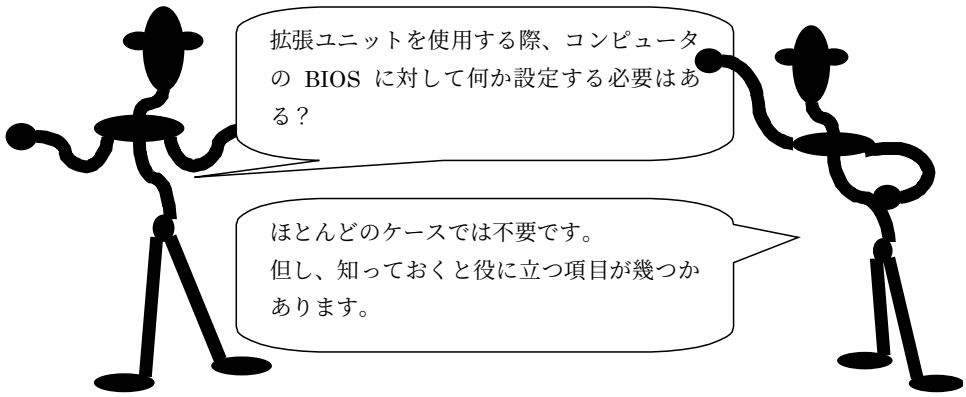
★何故、PCI-ISA および PCI-C(98)バスブリッジが使えないか？

PCI-ISA および PCI-C(98)バスブリッジは、製品の特性上 ISA や C(98)拡張ボードと同じ ISA 領域と呼ばれるメモリ(1MB 空間内)や I/O 領域と同じ領域を使用するため、これらのアドレスが使用可能な場所、即ちバス番号 0 に PCI ボードを挿して使用する必要があります。

★64 ビット PCI は使用可能か？

PCI は基本的に 32 ビットのバス規格ですが、高速化を施した 64 ビットのバス規格もあります。
この規格に沿って作られた 64 ビット PCI 拡張ボードを、弊社の拡張ユニットに挿して使用することは可能です。
但し、64 ビットでなく 32 ビットでの動作となります。

拡張ユニットを使用する際、BIOSに対して何か設定する必要はあるか？



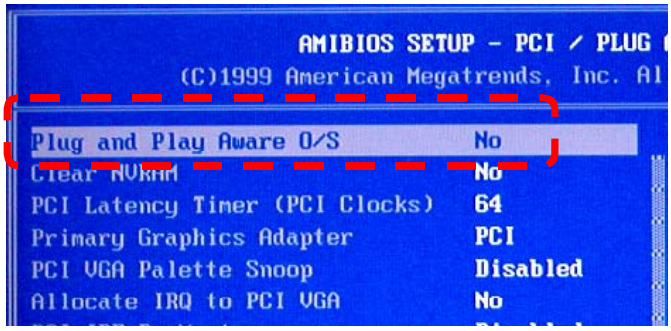
拡張ユニットを使用する際、コンピュータのBIOS設定で特別な設定が必要なケースは ほとんどありません。
但し、BIOS特有の動きについて、知っておくと いざという時、対処しやすいのも事実です。ここでは、BIOSのPCIに関する項目と動きについて幾つか紹介します。

★注意
ここで説明している機能は、コンピュータの BIOS によっては搭載されていないこともあります。
機能の搭載の有無は BIOS の優劣を決定するのでなく、リスク覚悟で細かいカスタマイズを許すか、お任せ設定を勧めるかの違いでしかありません。

Tips：OSの種類による設定

リソースの割り当ての再設定が可能なOSを使用する場合、BIOSでの設定を最小限にする方法があります。

下図は、弊社CPUボード製品PCI-PE09B3のBIOS設定画面の一部です。



ここの「Plug and Play Award O/S」(名称はBIOSによって異なります。例えば、「PNP OS Installed」など)を、Noに設定した場合、BIOSは全てのデバイスのリソースを設定します。Yesに設定した場合、BIOSは必要最小限の設定のみ行い、残りの設定をOSに任せようとします。
多くのケースでは、リソースの割り当てを自前で行うOSの場合、Yesに設定するのが望ましいです。

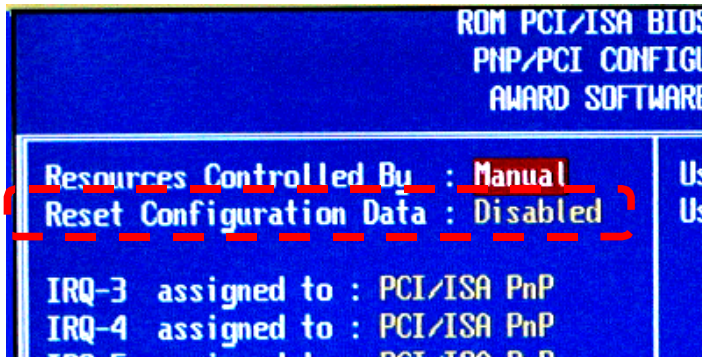
参考に、代表的なOSの推奨設定を紹介します。

設定値	推奨OS
Yes	Windows 98/Me Windows 2000/XP Linux
No	Windows 95/NT 4.0 MS-DOS PC DOS 2000

Tips：割り当ての再初期化

コンピュータに拡張ボードを挿したにも関わらずシステムが認識してくれない場合、BIOSに再初期化を指示する方法があります。

下図は、弊社CPUボード製品PCI-PE09B3のBIOS設定画面の一部です。


ROM PCI/ISA BIOS
PNP/PCI CONFIGU
AWARD SOFTWARE
Resources Controlled By : Manual
Reset Configuration Data : Disabled
IRQ-3 assigned to : PCI/ISA PnP
IRQ-4 assigned to : PCI/ISA PnP

ここの「Reset Configuration Data」をEnabledにした時、次回コンピュータ起動時、BIOSは内部に蓄えている情報を破棄し、一から再初期化を行います。
BIOSが蓄える情報は幾つかあります。デバイスの情報、レガシーカードについての情報、HDの情報、割り込みの設定などです。
(これらBIOSが蓄える情報をESCD(Extended System Configuration Data)と呼びます)

★初期化を確実に行わせるには
一般に、「Reset Configuration Data」項目を Enabled に設定し、コンピュータを再起動すれば初期化が行われますが、より確実に行わせるため、コンピュータに挿している拡張ボードのスロットの位置を一度変更することをお奨めします。

Tips：その他の設定

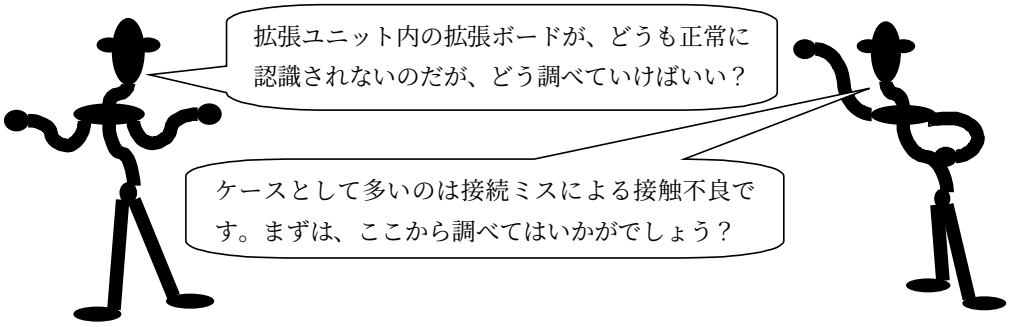
その他にBIOS設定で、PCIに関わるものとしては、IRQのルーティング設定と使わないデバイスのdisable設定が挙げられます。

IRQのルーティング設定は、PCIの割り込み信号からコンピュータへの割り込みコントローラへのルーティング設定を行うものです。
PCIの仕様に精通することが要求されるため、ここでは取り上げません。

使わないデバイスのdisable設定は、OSなどのシステムからデバイスを割り当てるだけのリソースが足りないと警告を受けた際、検討すべき項目です。
例えば、USBやLANなど、システム運用上 使用しないのであれば、これをdisable(使わない)設定にし、他のデバイスのリソース割り当て用に回します。

★切り離せないリソース
コンピュータによっては BIOS 上で無効に設定しても、デバイスを完全には切り離せないものがあります。この場合、リソース競合の要因となる場合もあります。
disable 設定を行う場合、デバイスの切り離しが正常に行われているか、様々な手段で確認を行います。

正常に認識されない時、どんな対処を行えば良い？



拡張ユニットに拡張ボードを挿しコンピュータと繋げた時、何故か正常に拡張ボードを認識できない場合があります。

一体何が問題で、正常に認識できないのか、弊社の技術サポートで承ったケースを元に幾つか事例を紹介します。

接続ミスによる問題

一番多いケースが、この接続ミスによるものです。

例えば、以下のようなケースです。

1. 拡張ボードが、ユニットに固定されていないなど、拡張スロットに十分挿しこまれてなかった。
2. 拡張ボードのカードエッジが汚れているため、接触不良を起こしていた。
3. (特に)PCI拡張ボードがガイドに固定されておらず、傾いて設置され、エッジが接触不良の状態になっていた。
4. バスケーブルが完全に挿し込まれてない。

拡張ボードが、拡張スロットに十分挿しこまれてなかったケースとは、例えば下図のような状態です。

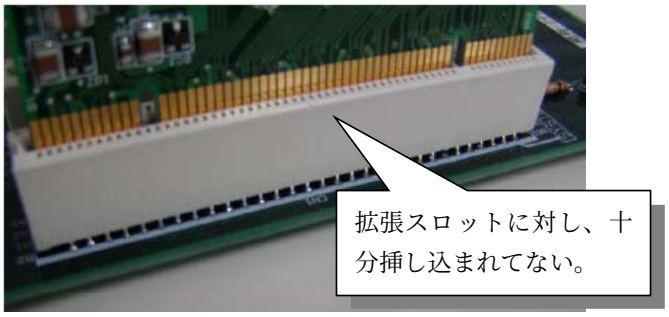


図 4-1 拡張ボードは拡張スロットに十分挿し込むこと

この状態は、拡張ボードのエッジがスロットにしっかり挿し込まれていないため、接触不良を起こしています。これでは正常に動作しません。

後、拡張ボードをスロットに挿入する際、カードエッジ部分が汚れてないことに注意してください。

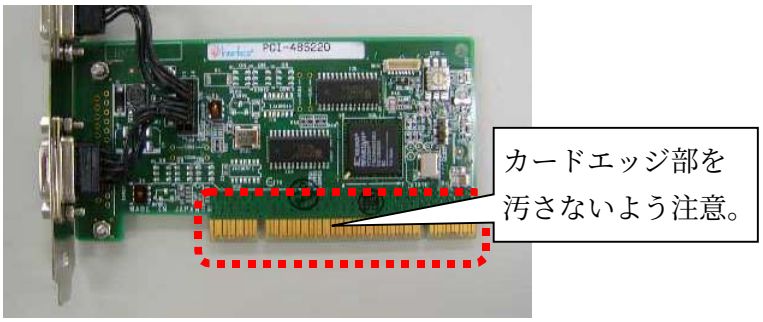


図 4-2 カードエッジ部が汚れないようにすること

拡張ボードを、しっかりスロットに挿し込んでも、カードエッジ部が汚れていると接触不良の原因となります。
カードエッジ部を直接手で持って、指紋などの汚れを付けてしまわないよう細心の注意を持って取り扱ってください。

次に、拡張ボードがユニットに固定されておらず、傾いて設置されていたため、エッジが接触不良のケースです。例えば下図のような状態です。

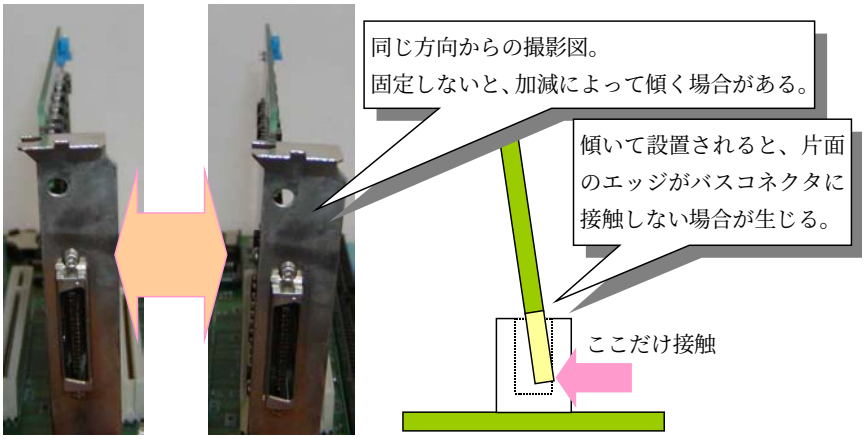


図 4-3 ユニットに固定しないと接触不良が起こることがある

これも接触不良の一つの例です。カードのエッジ部分はスロットの端子部分と十分接触する必要がありますが、ここでは傾いているため、片方の端子が十分接触できていません。
これを防ぐため、スロット上部にガイドがあり、ねじ止めにて固定することにより中立状態にさせ、十分端子を接触できるようになっています。

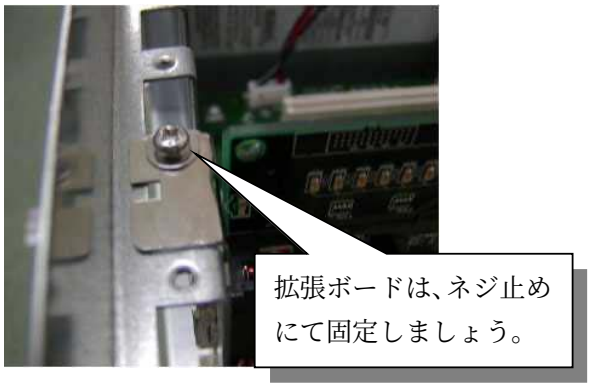
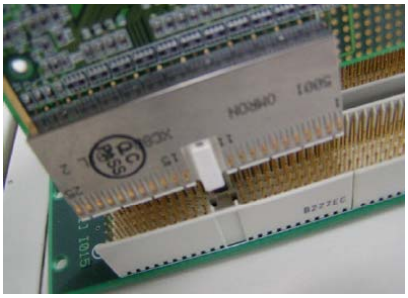


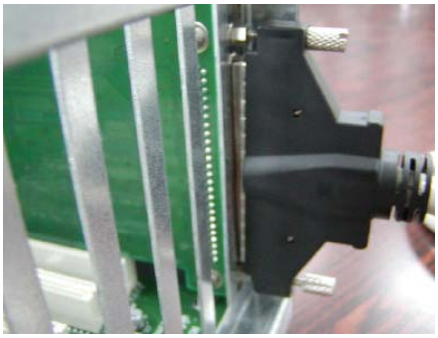
図 4-4- 拡張ボードは、ネジ止めで固定しましょう

★CompactPCI と PCI との違い
CompactPCI ボードの場合、PCI ボードに比べてバスコネクタがピンタイプなので、各ピンががっちり挟み込まれ、接触不良が起き難くなっています。
また、振動による抜けの恐れも、PCI ボードに比べて低くなっています。



次に、バスケーブルが挿し込まれてないケースは、例えば下図のような例です。

＜正しい接続例＞

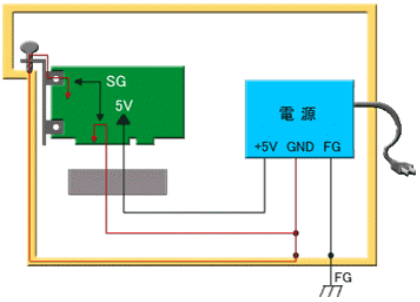

＜悪い接続例＞

このケースは、計測用の拡張ボードのコネクタにも言えることです。この場合は、ネジ付きならばネジ止めすることで、確実にコネクタを接続することができます。

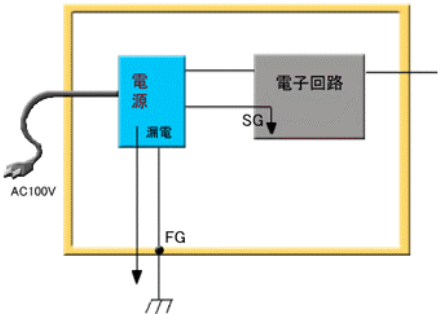
★拡張ボードやコネクタをネジ止めする その他の理由

拡張ボードやコネクタをネジ止めすることは、拡張ユニットに固定する以外にも様々なメリットがあります。
例えば、ノイズ対策としての効果や、リーク(漏れ)電流対策としての効果です。

ノイズ対策としての効果は、ユニットの筐体が SG(シグナルグランド)に接続されており、拡張ボードのパネル面も SG に接続されているため、双方をネジ止めすることで、ボード上のインピーダンスを下げ、安定した動作が期待できます。



リーク(漏れ)電流対策としての効果は、パネルが筐体に接触することで、リーク電流が筐体を通してグランドに流れ、ボードの誤動作を防ぐことができます。



このように、ネジ止めの効果は、様々な形で得られます。

特定のケースで、一部のOSあるいはBIOSがリソース設定できない

このケースは、CardBus－PCIバスブリッジ／CardBus－CompactPCIバスブリッジで起きる可能性のある問題です。

この問題で一番多いのが、一部のOSないしはBIOSが、CardBusバスブリッジの先のデバイスを正常に構成できない、あるいは構成方法をOSないしはBIOSが知らない、といったものです。

何故、一部のOSやBIOSで正常に構成できないのでしょうか？
これには、規格確定から広まるまでの時間的な兼ね合いや、CardBusバスブリッジ固有の特長による複合的な要素が噛み合わさっています。

例えば、CardBus規格はPCI規格が策定された後に作られましたが、CardBus製品が広まるまでの間出荷された、古いOSや古いBIOSを搭載したコンピュータでは、特定のケースで対応できない場合もあります。

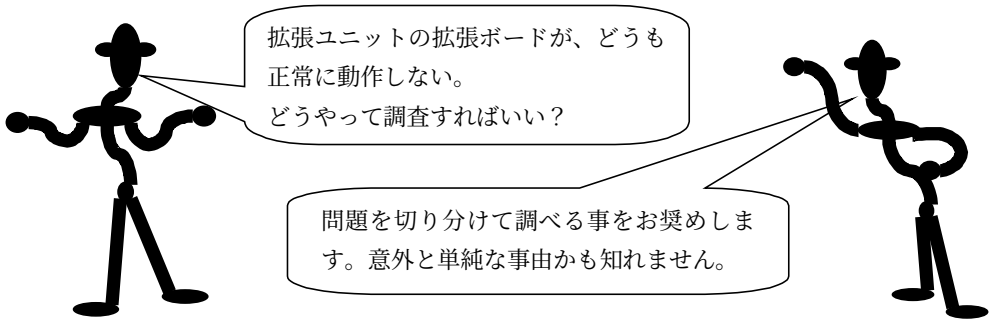
次に、CardBus固有の特長ですが、CardBusバスブリッジは、拡張ユニットに複数の拡張ボードを備えて動作します。このため、無線LANやGPSカードのような単一のCardBusカード製品に比べて、多くのハードウェアリソースを必要とします。これが、一部のOSやBIOSで予想外の処理となり、うまく構成できないことがあります。

対処方法としては、幾つかあります。

- (問題に対応している可能性のある)最新のOSを使用する。
- (問題に対応している可能性のある)最新のBIOSにアップデートする。
- 拡張ユニットに挿す拡張ボードの数を減らす。
- 弊社で公開している動作確認済みのコンピュータを使用して頂く。
- 弊社の貸出サービスをご利用頂き、購入前に動作確認をして頂く。
- 弊社で用意しているユーティリティを使い、構成の補助を行う。

弊社としては、より多くのお客様に満足してご利用頂くため、弊社の貸出サービスなどの様々なサービスにて事前確認をされることをお勧めします。

正常に動作しない時、どんな対処を行えば良い？



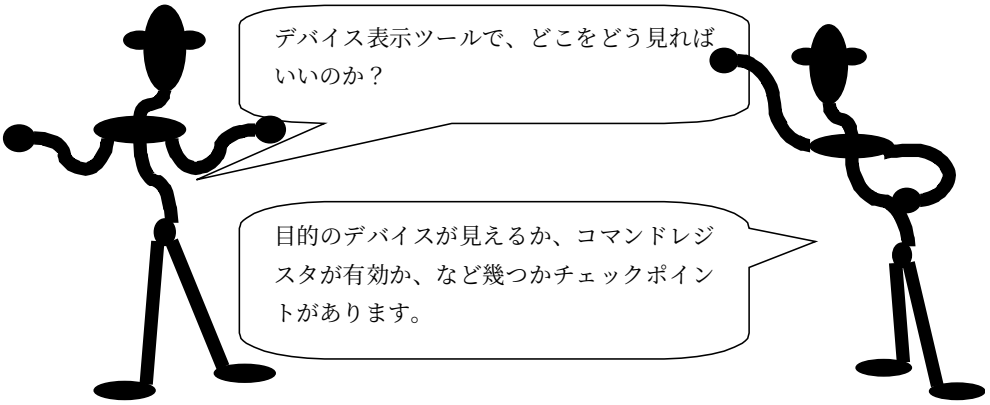
拡張ユニットに拡張ボードを挿しコンピュータと繋ぎ、デバイスマネージャ等で認識しているにも関わらず、どうも安定して動作しない場合どうすれば良いのでしょうか？

こういった問題に直面した時、一度基本に立ち戻ってチェックしてみると、意外な解法が見付かるものです。
ここでは、問題の切り分け方について、幾つかご紹介します。

対処	説明
問題を小分けにして確認	問題解決の一番の基本形です。 コンピュータと複数の拡張ボード、ソフトウェアを組み上げた複雑な状態で、原因を突き止めるのは非常に困難です。 複雑な現状のまま問題解決にあたると言うことは、いわば、数千ピースのジグソーパズルの山から、目的のピースを一気に探し出そうとしているのと同じことです。 大量のジグソーパズルを解く時、似たような絵柄のピースを選り出して、当てはめていきます。 システムの問題解決も同じ、問題を簡素化することから始めます。
OSが拡張ボードを認識しているか？	まずは、OSが拡張ボードを正常に認識しているか確認します。 弊社製品には、動作確認プログラムが大抵添付されておりますので、拡張ボードの動作に問題が発生していないかプログラムを使ってチェックするのも良いでしょう。 他にも弊社Web siteよりダウンロードできるデバイス表示ツールを使い、何か問題が発生していないか確認することもできます。

対処	説明
直接コンピュータのスロットにて動作確認する	拡張ユニット上の拡張ボードが動作するか、コンピュータの拡張スロットに直接拡張ボードを挿し、動作するか確認します。 コンピュータの拡張スロットで動作するなら、拡張ユニットに絡む問題であろうと予測できます。 逆に、コンピュータの拡張スロットで動作しないなら、拡張ボードかコンピュータの問題であると予測できます。
ドライバのバージョンを確認する	単体では動作するが、複雑な処理を行った時、どうも動作が異常の時、ドライバのバージョンが古くないか念のため確認しましょう。
拡張ボードの消費電流の合計を確認する	大量の拡張ボードや大きな消費電流を消費する拡張ボードを使用している場合は、消費電流の合計が拡張ユニットの供給する最大消費電流を下回っていることを確認します。 合計の消費電流が上回っているか、ほとんど同じ場合、更に大容量の拡張ユニットを用意すべきです。

問題が発生した時、デバイス表示ツールで、どこを確認すれば良い？



拡張ユニットや拡張ボードを使ったトラブルで、お客様から問い合わせがあった時、弊社ではよくお客様の環境下でのPCIコンフィギュレーション情報を送っていただき、原因の解析を行います。
デバイス表示ツールで、どこを確認すれば良いのか、簡単なチェック方法を紹介します。

チェックポイント１：目的のデバイスが見えるか？

下図は、デスクトップコンピュータにPCI-3161を挿し、デバイス表示ツールでPCIコンフィギュレーションレジスタの値を確認した画面です。

```
 PCI Device Viewer   Ver. 1.04 Copyright 2002 Interface Corporation.
       Bus  Dev  Func VendorID  DeviceID  RevisionID  ClassCode
  1:   00   00   00    8086      7120       03        000006
  2:   00   01   00    8086      7121       03        000003
  3:   00   1E   00    8086      2418       02        000406
  4:   00   1F   00    8086      2410       02        000106
  5:   00   1F   01    8086      2411       02        800101
  6:   00   1F   02    8086      2412       02        00030C
  7:   00   1F   03    8086      2413       02        00050C
  8:   01   02   00    1013      6003       01        000104
  9:   01   08   00    10B7      9200       74        000002
 10:   01   0A   00    1147      0C59       03        0000FF  <<<< Select
 11:
 12:
 13:
 14:
 15:
 16:
 Up/Down arrows: Select Device     Enter: Details     F3: Save     Esc: Exit
```

図 4-5 デバイス表示ツール：全体画面

まず、一番先に確認が必要なのは、目的の拡張ボード(PCIデバイス)が、デバイス表示ツールから参照できるか？という点です。

目的のPCIデバイスを見つけるには、「VendorID(ベンダーID)」と「DeviceID(デバイスID)」を参照します。
VendorIDは、デバイスのメーカを表します。この値を参照することで、PCIデバイスのメーカが判明します。弊社のVendorIDは1147hです。VendorIDが1147hのPCIデバイスがあれば、それは弊社のPCIボードであるとわかります。(余談ですが、8086hはIntelです)
DeviceIDは、デバイスを区別する一意のID値を表します。弊社では、製品の型式番号を基本的に割り当てています。

以上から、PCI-3161を探すには、VendorIDが1147h(インタフェース社を表す)で、DeviceIDが0C59h(10進数に直すと3161。PCI-3161の型式番号を指す)のデバイスを、PCIデバイス表示ツールから探し出せば良いことがわかります。

上図では、10番目のPCIデバイスが該当します。

ここで目的のPCIデバイスが見付からなかった場合は何を意味するかと言うと、「コンピュータがデバイスを論理的に参照できていない状態にある」を意味します。
つまり、拡張ボードがスロットにうまく挿し込まれておらず、接触不良の状態なため、認識できてないとか、接続ケーブルが抜けかかっているなどです。

★接触不良によるビット欠け
接触不良により中途半端にデバイスを参照できる状態になることがあります。
そんな時、VendorID や DeviceID の値がビット欠けの状態で参照できることがあります。

例えば、PCI-3161 の DeviceID は 0C59h ですが、bit0 が欠けて 0C58h として表示されるなどです。
VendorID が 1147h(弊社)のボードなのに、見かけない DeviceID を持つデバイスが見付かった時、こういった接触不良を疑ってみるのも良いかも知れません。

チェックポイント２：コマンドレジスタは有効か？

PCIデバイス表示ツールで、コンピュータからデバイスが参照できることを確認したら、次は、デバイスがコンピュータによって正しく構成されているか、確認する必要があります。

下図は、PCIデバイス表示ツールで、先の10番目PCI-3161のデバイス情報を詳細に表示させた所です。

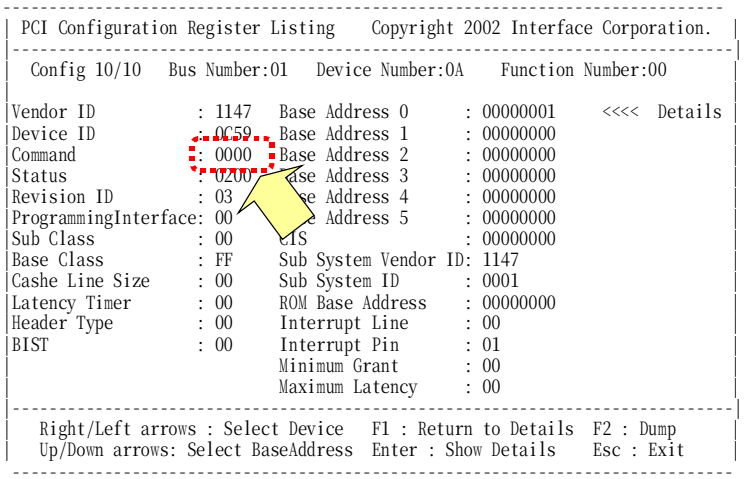


図 4-6 デバイス表示ツール：詳細画面

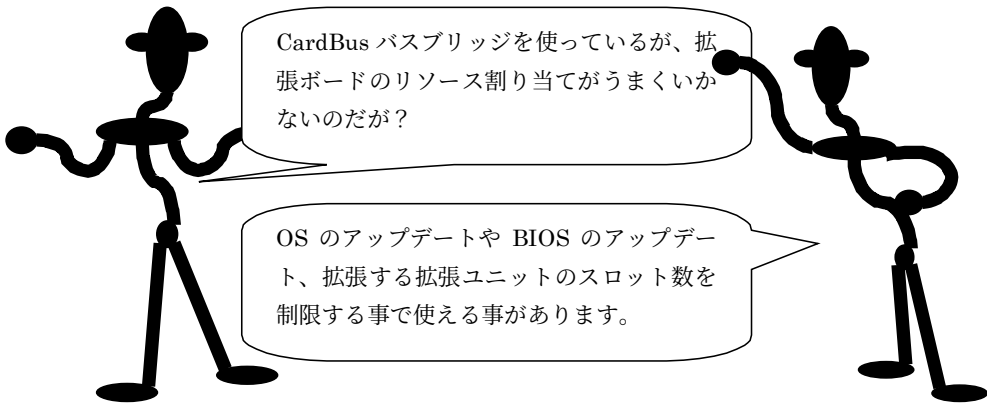
ここでは、「Command」の値をチェックします。Commandは、PCIデバイス各機能を保持する重要なレジスタです。各ビットの割り当て意味を少し紹介します。

ビット	説明
0	I/O 空間制御 0:デバイスによる I/O 空間アクセスへの応答を禁止します。 1:デバイスによる I/O 空間アクセスへの応答を許可します。
1	メモリ空間制御 0:デバイスによるメモリ空間アクセスへの応答を禁止します。 1:デバイスによるメモリ空間アクセスへの応答を許可します。
2	バスマスタ制御 0:デバイスの PCI バスマスタアクセスを禁止します。 1:デバイスの PCI バスマスタアクセスを許可します。
…	…

例えば、PCI-3161は、I/Oポートとメモリアドレスを使用するボードです。
ここで、ビット0とビット1が0ということは、PCI-3161のI/Oポートとメモリアドレスが使用できない状態にあるということです。

BIOSやOSが正常にPCIデバイスを構成できなかった時、Commandの値は0クリアされたりします。
この値を確認することで、PCIデバイスがコンピュータからアクセスできる状態にあるか確認することができます。

CardBusバスブリッジで、デバイスのリソースが割り当てられない



『特定のケースで、一部のOSあるいはBIOSがリソース設定できない』にて説明したように、CardBusバスブリッジを使用して拡張ボードを使用する時、特定のケースで正常に使えないことがあります。

一番良い方法は、CardBusバスブリッジが使えるノートパソコンで ご使用いただくことですが、現在お使いのノートパソコンで使用できるようにするためのTipsを幾つかご紹介します。

Tips：最新のOSとアップデートを施す

最新のOSとアップデートを施す理由は、CardBusに対するOS側の対処や修正が施されている可能性が高いからです。

弊社では、最新のサービスパックがインストールされていないOSで拡張ボードの数が多くなると、リソース割り当ての最適化等が行われず、認識されない問題が発生したことを確認しています。

Tips：少ないスロット数の拡張ユニットを使用する

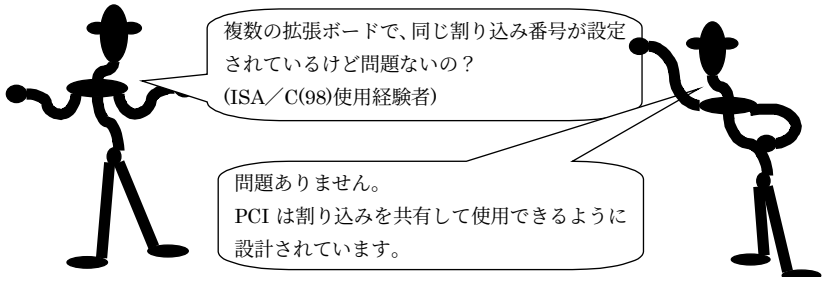
スロット数の多い拡張ユニットは、PCI規格の制約上 スロット数の少ない拡張ユニットに比べて 多くのPCI-to-PCIブリッジを内蔵しています。

WindowsなどのOSはリソース割り当て時、PCI-to-PCIブリッジを見つけると、その先にあるであろうPCIデバイスのために、ある程度のリソースを確保しようとします。
つまり、PCI-to-PCIブリッジが増えるほど予め要求されるリソースが大きくなることになります。

例として、スロット数の異なる拡張ユニットでCardBusバスブリッジの最小使用リソース量を比較しましょう。
(データは弊社環境で確認したものです。お客様の環境では異なる結果が得られる可能性があります)

種類	使用するI/Oポート (最小)	使用するメモリ (最小)
CardBus-PCI(4スロット)	300h	200000h
CardBus-PCI(7スロット)	2000h	200000h
CardBus-PCI(13スロット)	4000h	4000000h
CardBus-CompactPCI(4スロット)	1000h	200000h
CardBus-CompactPCI(7スロット)	1000h	200000h
CardBus-CompactPCI(13スロット)	2000h	4000000h

拡張ボードの割り込み番号が重複しているが、使用しても大丈夫か？

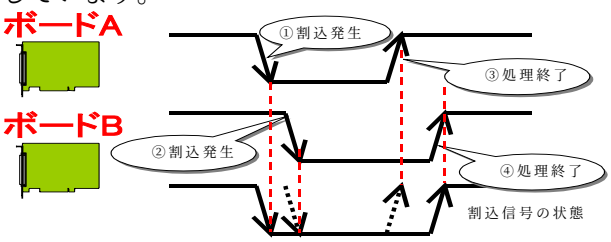


Windowsのデバイスマネージャなどで PCIデバイスの割り込み番号を見ると、複数の拡張ボードが同じ割り込み番号を割り当てているのに気付きます。
ISAやC(98)拡張ボードを使用されていた方にとって、拡張ボード1枚ごとに割り込み番号を変えることは絶対だったため、果たして大丈夫か？と心配される場合もあるでしょう。

PCIでは割り込みの処理方法をエッジトリガ割り込みからレベルトリガ割り込みに変更したため、複数の拡張ボードで割り込みを共有(シェア)することができるようになっています。

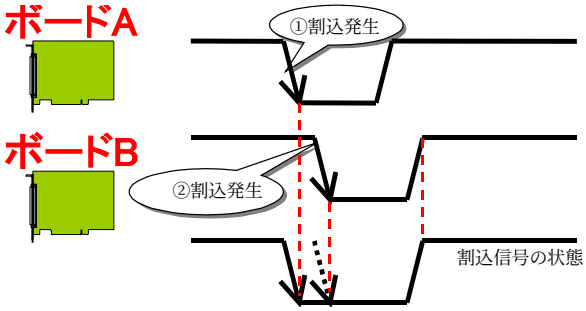
★何故、PCI で割り込みが共有できるのか？
エッジ割り込みからレベルトリガ割り込みにしただけで、何故 PCI で割り込みが共有できるのか不思議に思われるかも知れません。
これは複数の割り込みが同時に入った状況を考えると納得できます。

下図は A と B のボードが ほぼ同時に割り込みをコンピュータに対して通知した状況を現しています。



まず、①と②でボード A とボード B が割り込み信号を”Low”レベルにドライブして、割り込みを要求します。
コンピュータは割り込み信号が”Low”レベルを検知し、ボード A かボード B のどちらかで割り込みが発生したことを知ることができます。
コンピュータはボード A の割り込み処理を終え、③でボード A は”Low”レベルからハイインピーダンス状態に戻します。しかし、ボード B は割り込み処理が終わっていないので、”Low”レベルにドライブしたままです。このため、割り込み信号は”Low”レベルを維持します。
これにより、コンピュータは他に割り込みを行っているボードが居ることを知り、次にボード B の割り込み処理を行います。
ボード B の割り込みを完了して両方のボードの割り込みが完了したため、④で割り込み信号が”Low”レベルから”High”レベルに戻ります。

これがエッジトリガ割り込みだとどうでしょうか？



①と②でボード A とボード B が割り込み信号を”Low”レベルにドライブする所までは同じです。
しかし、エッジ割り込みのため、一定時間が経つとボード A、ボード B 共、信号を”Low”レベルから”High”レベルに戻します。このため、コンピュータから見て割り込みが入ったことは判りますが、入った割り込みは片方だけなのか両方なのか判断する術がありません。(PCI では割り込みが完了しないデバイスが割り込み信号を”Low”レベルにドライブしたままだという点に注目してください)
これが PCI で割り込みを共有できる仕掛けです。

このようにPCIでは割り込みを共有できる訳ですが、一点問題があります。
それは、割り込みの共有を実現するには、ハードウェアとソフトウェアの対応が揃っている必要があるということです。つまり片方だけ動いても、もう片方が正しく動かない限り、割り込みの共有は実現できません。

両方が揃ってないと、例えば以下のようなケースが挙げられます。

1. 割り込みを共有していない状態では正常に動作するのに、割り込みを共有したら、割り込み処理が必要なアクセス時にハングアップする。
2. 割り込みを共有したら全くデータを取り込まなくなった。
3. ある時点からアクセスLEDが点きっ放しの状態になって固まる。

これらのケースのほとんどがソフトウェア側の問題に起因するものです。

1. 確認および対処方法としては、以下が考えられます。
   割り込みを共有しない状態でテストする。
   Windowsならデバイスマネージャの設定で変えられる場合があります。または、拡張ボードを挿すスロットを変更することやBIOSの設定で変えられる場合もあります。

2. ソフトウェア側の問題と判明したら。
   問題が判明した際の対処としては、割り込みを共有しない設定にするのと、ソフトウェアの修正対応を行う、の選択が考えられます。
   PCIでは仕様上ハードウェアリソースを固定しにくいため、ソフトウェアの修正対応を行うのがより望ましいと言えます。

用語集

ここでは、拡張ユニット関連の用語について説明します。

用語	説明
拡張ユニット	PCI/CompactPCIバスを拡張するためのユニット。 弊社の拡張ユニットを使用することで、パソコンの空きスロットが足りない場合でも4スロット、7スロット、13スロットの拡張ができる。 また、ノートパソコン(32ビットCardBus)や、Low Profile PCI搭載パソコンからもPCI/CompactPCIバスを使用することができる。
高速シリアルバスブリッジ	弊社の高速シリアルバスブリッジは、データ転送を確保しやすいネットワーク規格であるStarFabricを採用している。従来のバスブリッジ製品と違い、最大12mまで延長できる。
StarFabric	Star Fabricは、データ転送速度を確保しやすいスイッチド・ファブリック技術を基盤技術としたネットワーク規格である。 PCIバス・システムなどの既存システムの拡張性と信頼性を低コストで実現できる。 既存のPCIバス・システムのハードウエアとソフトウエアを変更せずに、複数のPCIバスを相互に接続できる。
拡張スロット	コンピュータの機能を拡張するボードやカードを搭載するための差し込み口(スロット)のこと。拡張バス・スロットとも呼ぶ。バスの種類によってその形状や接点数は異なる。拡張スロットの数が多いほど、一般的に拡張性は高くなる。
拡張バス	拡張スロットとCPUを結ぶ外部バスのこと。
ラックマウント	ラック(rack)は、棚、架という意味。パソコンラックという言葉もある。マウント(mount)は、ものを何かに据え付けたり、はめ込むという意味。 「ラックマウント型のXX」あるいは「ラック型のXX」というのは、「専用のラックにマウントするタイプ」という意味になる。 ラックの幅は、EIA(Electronic Industries Alliance=米国電子工業会)やJISにより、19インチに統一されている。このラックを、キャビネットということもある。 ※CompactPCIのユニットをラックに取り付けるための専用金具も用意してあります。
ラックユニット	棚状の台にボードやユニットを組み合わせて一つのシステムを作ったもの。
電源連動制御機能	ホスト側コンピュータの電源のON/OFF、またはカードの挿抜に応じて自動的に拡張ユニットの電源がON/OFFできる機能のこと。

技術資料紹介

弊社では下記の**技術資料**を提供しております。
詳しくは、弊社Web site(www.interface.co.jp)、または弊社窓口までお問い合わせ下さい。

カタログ	
PRM-0061	CPZカタログ(日本語版)
PRM-0062	PCIカタログ(日本語版)
PRM-0063	CSIカタログ(日本語版)
チュートリアル	
TUT-0056	チュートリアル　XP Embedded OS構築編
TUT-0055	チュートリアル 画像入力ボード
TUT-0054	CANチュートリアル
TUT-0053	モーションコントロールチュートリアル
TUT-0050	RTLinuxによるモーションコントローラボード制御プログラミング チュートリアル(GPG-7400用)
TUT-0048	RTLinuxによるカウンタボード制御プログラミング チュートリアル
TUT-0044	RTLinuxによるメモリ共有インタフェースボード制御プログラミング チュートリアル
TUT-0043	RTLinuxによる調歩同期シリアル通信ボード制御プログラミング チュートリアル
TUT-0041	RTLinuxによるGP-IBボード制御プログラミング チュートリアル
TUT-0040	RTLinuxによるDAボード制御プログラミング チュートリアル
TUT-0039	RTLinuxによるADボード制御プログラミング チュートリアル
TUT-0038	RTLinuxによるDIOボード制御プログラミング チュ-トリアル
TUT-0037	RTLinuxによるHDLCボード制御プログラミング チュートリアル
TUT-0036	RTLinuxによるPCI/CompactPCI/CardBus制御入門書(導入編)
TUT-0034	Visual C++によるPPI入門書
TUT-0033	Visual Basicによるメモリ共有インタフェース入門書
TUT-0032	Visual C++によるメモリ共有インタフェース入門書
TUT-0031	Visual Basicによるカウンタ入門書
TUT-0030	Visual C++によるカウンタ入門書
TUT-0029	Visual BasicによるHDLC入門書
TUT-0028	Visual C++によるHDLC入門書
TUT-0027	Visual BasicによるGP-IB入門書
TUT-0026	Visual C++によるGP-IB入門書
TUT-0025	Visual BasicによるDIO入門書
TUT-0024	Visual C++によるDIO入門書
TUT-0023	Visual BasicによるDA入門書
TUT-0022	Visual C++によるDA入門書
TUT-0021	Visual BasicによるAD入門書
TUT-0020	Visual C++によるAD入門書
TUT-0019	Visual Basicによるモーションコントローラ入門書
TUT-0018	Visual C++によるモーションコントローラ入門書
TUT-0017	メモリンクを使用した負荷分散システム事例チュートリアル
TUT-0016	Visual BasicによるPPI入門書
TUT-0015	モーションコントロールチュートリアル
TUT-0014	Microsoft Visual Studio .NET移行ガイド
TUT-0008	拡張ユニット チュートリアル(問題解決編)
TUT-0007	拡張ユニットチュートリアル(入門編)
TUT-0006	C(98)/ISA製品からPCI/CompactPCI製品への移行チュートリアル(DOS編)
TUT-0005	DOSによるLAP-B入門書
TUT-0004	DOSによるAD入門書
TUT-0003	LinuxによるPCI/CompactPCI/CardBus制御 入門書
TUT-0002	PCI-ISAバスブリッジチュートリアル
TUT-0001	PCI-Cバスブリッジチュートリアル
技術情報資料	
	初めてのCANインタフェース
	Linux，リアルタイムLinux移植(SH-4)経験談及び当社の今後の取り組みについて
	LinuxからPCI/CompactPCIボードを制御する方法
	ActiveXコントロールによるシステム組み込み技術
	CompactPCIへの置き換え+システム構築/移行ガイド
	MS-DOSからPCI/CompactPCIボードを制御する方法

参考文献

大川 善邦 著：PCIバスによるI/O制御(オーム社：1999年)

Edward Solari、George Willse 著：PCIハードウェアとソフトウェア(インフォ・クリエイツ：1995年)

拡張ユニット チュートリアル(問題解決編)

2005年 9月　Ver. 1.1　発行
発行所
Dream & Freedom 株式会社 インタフェース

〒732-0828
広島県広島市南区京橋町10-21
TEL　082-262-7777
FAX　082-262-5066

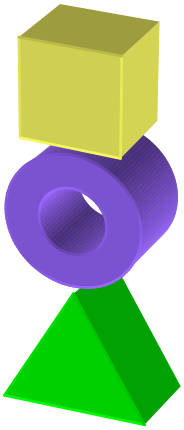
ISD-7080-11
価格 ¥2,100 (**定価 ¥2,000** + 消費税 ¥100)

本書の内容の一部または全部を、無断で転載することを禁止します。
本書の内容は、将来予告なく変更することがありますので、あらかじめご了承ください。© 2003, 2005 Interface Corporation. All rights reserved.

Interface®

拡張ユニットチュートリアル（問題解決編）

TUT-0008　Ver. 1.1



Interface®

www.interface.co.jp